

IS THERE REALLY TROUBLE WITH UNIX* ?

Lorenzo De Leon
Teletype Corporation
Skokie, Illinois 60077

William G. Harris
Psych Systems Inc.
Baltimore, MD 21208

Martha Evens
Department of Computer Science
Illinois Institute of Technology
Chicago, Ill. 60616

ABSTRACT

Donald Norman has claimed that UNIX has cryptic and inconsistent command names. As Michael Lesk has remarked, the lack of objective data makes it difficult to evaluate the significance of Norman's criticisms. In an effort to explore this controversy we taught one group of novice users the UNIX command language and another group an English-based command language (NUIX). The number of errors and calls for on-line assistance were compared. The subjects in this study were 22 high school women with no formal exposure to computers. The results reveal that the UNIX group made fewer errors than the NUIX group in two training sessions a week apart. Although calls for on-line assistance for the two groups in the first session were comparable, the UNIX group made over twice as many calls for on-line assistance in the second session as the NUIX group. Our findings suggest that even though the UNIX command language may not be harder for novice users to learn, it is probably more difficult for them to use.

INTRODUCTION

Although UNIX is an elegant and powerful operating system, it is embroiled in a controversy over its human interface. Donald Norman asserts in his article "The Trouble with UNIX" [Datamation, November 1981, 139-150], that there are serious weaknesses in the UNIX user interface. For example, command names are often inconsistent and are not easily related to their respective functions. In his rebuttal, Michael Lesk from the Bell Telephone Laboratories points out that Norman has no experimental evidence to support his thesis. The main objective of this paper is to investigate user problems with the UNIX command names in an effort to explore this controversy further.

Many good things have been said about the UNIX operating system developed by the Bell Telephone Laboratories and deservedly so. It has a sleek system design that includes a hierarchical file structure that has become the norm for modern operating systems. UNIX also offers input and output redirection, and a powerful command language called the shell. These and other features of UNIX have led to its wide acceptance and to its present status as an operating system standard.

Although Norman has criticized several aspects of the UNIX command language, our investigation focuses on the issue of cryptic command names. In particular, are these cryptic command names difficult for novice users to learn and understand? We chose to examine the command name issue for two reasons. First, it is the logical starting point when considering the human aspects of a command language. Second, this issue of cryptic command names lends itself to empirical evaluation.

THE SAMPLE

Subjects in this study were high school women who had voluntarily signed up for three separate computer literacy workshops. The workshops were sponsored by the Women In Engineering and Science program (WIES) at the Illinois Institute of Technology. The age of the subjects ranged from fourteen to seventeen years. Each workshop consisted of subjects from a different age group. The first workshop was made up of students in the 11th and 12th grades. The second workshop included students from the 10th and 11th grades and students from the 9th and 10th grades composed the third workshop. Overall the students had performed above average in their course work in science and mathematics. (Consult Table 1.)

Students with a background in computers were not allowed to register for the workshops. This

* UNIX is a registered trademark of the Bell Telephone Laboratories.

ensured a population of novice computer users. A novice computer user is defined as a subject with no formal exposure to computers or computer languages. The use of novice users differs slightly from Norman's focus on the casual user. A casual user is defined as an occasional user of computers. The decision to use novice users was made on the premise that novice users were more clearly defined and more readily available.

THE HARDWARE ENVIRONMENT

The hardware environment consisted of a DEC VAX 11/780 operating under the VMS operating system. All subjects used interactive CRT terminals that were connected to the VAX through a port contender. The terminals were located in a terminal room containing approximately 25 terminals. Each student was stationed at her own terminal.

THE SOFTWARE ENVIRONMENT

We initially designed a computer literacy workshop for the WIES program, and developed a series of laboratory exercises for the workshop participants. We determined a small subset of twelve commands that were needed to execute these exercises. Since the VAX was not running the UNIX operating system, the UNIX shell was not resident. A simulated UNIX shell that contained the subset of twelve commands was developed. This pseudo-shell intercepted UNIX shell commands from the user and generated the proper DCL (Digital Command Language) commands needed for execution. For example, a "MV" command (which renames a file) would generate a DCL "RENAME" command.

Since the focus of the investigation was to study command names, we wanted to isolate that effect from other possible influences. We felt that the entering of parameters on a command line was such an influence. In order to rename a file in the UNIX command language for example, two parameters are needed in a specific order; the original file name followed by the new file name. We felt that requiring the user to remember both the command names and their respective parameters would introduce an unnecessary variable to our study. The question of how command parameters affect the user is a very important topic and should be addressed in its own right.

However in the present study we tried to negate the effect of entering parameters by making the process interactive. In the experiment, after the user entered a command, specific parameters were individually requested. For example, in the renaming of a file the user was requested to enter the original file name and the new file name. This removed from the user the burden of remembering which commands required which parameters in what order.

To facilitate the learning of the command language, an on-line assistance feature analogous to the UNIX "MAN" command was implemented. When this command was executed, a list of the commands and their respective functions was displayed on

the user's screen. The screen was cleared before the next command was entered to prevent the user from being overly dependent on this list.

A duplicate of this UNIX pseudo-shell with a different set of command names was also created. This second command language functioned exactly like the first except with different command names. The command names were actual English words that described the function of the command. This second command language was called the NUIX command language. The two command name lists are given in Table 2.

Along with these two pseudo-shells, a user monitor was also developed. This monitor recorded every command that was entered by the user and marked those that were in error. This gave us the facility to record user error rates and the number of calls to the on-line assistance command.

THE WORKSHOPS

There were three computer literacy workshops. Each workshop consisted of three sessions which were held on three successive Saturdays.

During the morning of the first session, the students were introduced to the WIES program and to engineering in general. In the afternoon, the students were shown a film on computers and were given a brief introduction to computers and command languages. At this point, the students were randomly divided into two groups. Each group was then taught a different command language by a different laboratory assistant. One group was taught the UNIX command language while the other group was taught the NUIX command language. These lectures were both about forty-five minutes long. All the students were instructed not to take notes. After the lecture, both groups were led into the terminal room where each student was seated at her own terminal near other members of her group. The intent was to separate the groups to prevent confusion. The instructor stood in front of the room and assigned a series of computer tasks to the students. These tasks included such activities as copying, renaming, comparing, and displaying files as well as all the other commands that were contained in both of the command languages. The students were not allowed to use any materials to assist them with their exercises. Whenever a lab assistant was approached by a student with a request for help, the assistant directed her to the on-line assistance instead of answering the question directly. This first terminal session lasted approximately an hour and served as an introduction to command language usage for the students. After this terminal session, the students were asked to rate their individual performance using their respective command language.

The first activity of the second Saturday was another computer session. This session lasted approximately one half hour. After the terminal session the students were given a tour of the I.I.T. computer facilities and were involved in a

discussion of computer professions. The second afternoon ended with another half hour terminal session. The students were again asked to rate their individual performance.

The third morning started with the last terminal session. During this session, the on-line assistance commands were disabled. This was done in order to test the students' recall of their command language. The students were asked to experiment if they could not remember a particular command. Afterwards the students were again asked to rate their performance and were additionally asked to rate the ease of use of their respective command language. A conversation with a notable woman software engineer ended the morning's schedule. The afternoon was occupied by various WIES activities.

THE RESULTS

The experimental framework was such that subjects were compared under different conditions, therefore an analysis of variance with repeated measures was used. All analyses were based on the subject's error and help rates. The subject's error rate was computed by dividing the number of erroneous commands by the total number of commands. The subject's help rate was calculated by dividing the calls to on-line assistance by the total number of commands.

Table 3 contains the means and standard deviations for group error and help rates for sessions one and two. As shown in this table, the error rate for the NUIX group dropped from 7.2 to 3.6. The error rate for the UNIX group had a lower initial value of 3.8 and dropped only slightly. The help rate for groups was about the same in session one, but the NUIX help rate was less than half of the UNIX help rate in the second session. The main effect for error rate between the NUIX and UNIX groups was significant at the .05 level ($F(1,22) = 4.97$). Similarly we found a significant effect between sessions at the .05 level ($F(1,22) = 5.57$). A non-significant main effect for help rate for language groups was found. However, a significant main effect for sessions was found at the .01 level ($F(1,22) = 12.70$). We also found a significant interaction between groups and sessions ($F(1,22) = 5.30$). A significant difference in help rates between NUIX and UNIX in session two was found. A t-test comparing the means resulted in a significance to the .01 level ($t = 3.182$).

Table 4 contains only the group error rates for the third session because the on-line assistance was disabled during this session. The NUIX error rate was slightly more than half of the UNIX error rate for this session. A t-test analysis of the variance of the error rate did not show a significant difference.

DISCUSSION AND CONCLUSIONS

This experiment was designed to investigate the effect of command names on novice users. Recall of command names was tested with and

without on-line assistance. We found that the UNIX group had a lower error rate with the on-line assistance enabled. When the on-line assistance was disabled, the UNIX group seemingly had a larger error rate. This result however was not statistically significant.

If we use the help rate as an additional measure of recall, we see that while both groups had the same initial help rate, the NUIX group had a significantly lower help rate in the second session. The UNIX group continued to need a substantial amount of help.

Some of these results seem counter-intuitive. We expected a command language based on English words to have fared better than the UNIX command language. We have come up with several possible explanations for our results. One possibility is that the familiar nature of the English command names of the NUIX command language may have promoted overconfidence in the users. Another related possibility is that synonyms may have been confused with the English-based NUIX commands. For example, the words "FIND" and "SEEK" may have substituted for the NUIX command "SEARCH". In contrast, the UNIX command "GREP" probably would not have such word associations. A third possibility is that subjects unfamiliar with terminal keyboards might have made more typing errors with the longer NUIX commands. A fourth possibility is that novices may associate new words with the new concepts they have learned more easily than attaching new meanings to already familiar words.

In order to investigate these possibilities further, we are currently examining the actual individual errors that were made by the subjects. Because the last workshop concluded one week before the deadline for this paper, we will present these results later.

Our findings show that novice users have less of a reliance on on-line assistance with an English-based command language than with the UNIX command language. This is evident when we view the results from the first two sessions. Although session three did not yield a significantly different error rate, it did produce a need for further investigation. The findings of our study are not directly generalizable to Norman's criticisms of the UNIX command language. This is because we studied novice users while Norman was concerned with casual users. Our findings do suggest that even though the UNIX command language may not be harder for novice users to learn, it is probably more difficult for them to remember.

Table 1.

Demographic Characteristics of NUIX and UNIX subjects.

	NUIX	UNIX
N	11	13
Mean Age	15.15 (0.5)*	15.4 (0.9)
Mean G.P.A. (4 point system)	3.2 (0.6)	3.5 (0.5)
Mean Math G.P.A.	3.3 (0.4)	3.5 (0.5)
Mean Science G.P.A.	3.5 (0.7)	3.5 (0.5)

* Enclosed value is standard deviation.

Table 2.

Command Names and Functions.

NUIX	UNIX	FUNCTION
HELP	MAN	ON-LINE ASSISTANCE
DIRECTORY	LS	LIST CONTENTS OF DIRECTORY
DISPLAY	CAT	DISPLAY CONTENTS OF FILE
DELETE	RM	DELETE FILE
RENAME	MV	CHANGE NAME OF FILE
COPY	CP	MAKE COPY OF FILE
COMPARE	CMP	COMPARE FILES
SEARCH	GREP	SEARCH FOR STRING IN FILE
USERS	WHO	LIST SYSTEM USERS
TERMINAL	TTY	DISPLAY TERMINAL CHARACTERISTICS

Table 3.

Mean and Standard Deviation of Group Error and Help Rates
for Sessions 1 and 2.

	Session 1		Session 2	
	Error Rate	Help Rate	Error Rate	Help Rate
NUIX	7.2 (5.2)*	31.7 (10.8)	3.6 (2.5)	11.6 (9.5)
UNIX	3.8 (3.1)	32.0 (15.4)	2.3 (2.5)	26.9 (13.9)

* Enclosed value is standard deviation.

Table 4.

Mean and Standard Deviation of Group Error Rate
for Session 3.

	Error Rate
NUIX	10.1 (9.1)*
UNIX	19.4 (19.1)

* Enclosed value is standard deviation.